

CLAIMS

We Claim:

1. A method of debugging an electronic design comprising:
 - identifying a source of an internal signal of said electronic design to be viewed;
 - identifying an output pin of said electronic design to which it is desired to route said internal signal;
 - receiving a number indicating a quantity of registers to be inserted between said internal signal source and said output pin;
 - performing a compile of said electronic design including compiling a routing from said internal signal source to said output pin via said quantity of registers; and
 - producing an output file representing said electronic design as a result of said compile, said output file including an additional delay from said internal signal source to said output pin due to said quantity of registers.
2. A method as recited in claim 1 wherein said compile is an incremental recompilation, said method further comprising:
 - performing a full compile of said electronic design prior to said incremental recompilation, wherein said incremental recompilation does not disturb the placement and routing resulting from said full compile.
3. A method as recited in claim 1 wherein said quantity of registers is received via user input.
4. A method as recited in claim 1 wherein said quantity of registers is received via software that automatically calculates said number.

5. A method as recited in claim 1 further comprising:

identifying a clock signal to be used to trigger said registers.
6. A method as recited in claim 1 wherein said registers are flip flops, latches or more complex registers.
7. A method as recited in claim 1 further comprising:

downloading said output file onto a PLD, whereby said compiled design may be tested on said PLD and said internal signal viewed at said output pin.
8. A method as recited in claim 1 wherein the recited steps of claim 1 are performed in a system using an electronic design automation (EDA) tool.
9. A method as recited in claim 1 wherein said electronic design is intended for a programmable logic device (PLD), a chip, or a circuit board.
10. A method of debugging an electronic design comprising:

identifying a plurality of internal signals of said design to be viewed, each of said internal signals having a source;

identifying a plurality of output pins of said design to which it is desired to route said internal signals, each of said signals being routed to one of said output pins;

for each internal signal, receiving a number indicating a quantity of registers to be inserted between the internal signal source and its corresponding output pin;

performing a compile of said electronic design including compiling a routing, for each internal signal, from said internal signal source to its corresponding output pin via said registers corresponding to said internal signal; and

producing an output file representing said electronic design as a result of said compile, said routing of said output file arranged to synchronize said internal signals at said output pins due to said registers.

11. A method as recited in claim 10 wherein said compile is an incremental recompilation, said method further comprising:

performing a full compile of said electronic design prior to said incremental recompilation, wherein said incremental recompilation does not disturb the placement and routing resulting from said full compile.

12. A method as recited in claim 10 wherein said quantity of registers for each internal signal is received via user input.

13. A method as recited in claim 10 wherein said quantity of registers for each internal signal is received via software that automatically calculates said number.

14. A method as recited in claim 10 further comprising:

for each internal signal, identifying a clock signal to be used to trigger said registers corresponding to said internal signal.

15. A method as recited in claim 10 wherein said registers are flip flops, latches or more complex registers.

16. A method as recited in claim 10 further comprising:

downloading said output file onto a PLD, wherein said compiled design is tested on said PLD and wherein said internal signals are synchronized at said output pins.

17. A method as recited in claim 10 wherein the recited steps of claim 10 are performed in a system using an electronic design automation (EDA) tool.

18. A method as recited in claim 10 wherein said electronic design is intended for a programmable logic device (PLD), a chip, or a circuit board.

19. A method as recited in claim 10 wherein said internal signals form a bus of said electronic design.

20. A method of performing an incremental recompile of an electronic design comprising:

skipping the database building and logic synthesis stages of a full compilation;

retrieving a logical and a routing netlist for said electronic design;

receiving an internal signal name and an output pin name;

creating a number of registers to insert between said internal signal and said output pin;

connecting said internal signal to said output pin via said registers in said logical netlist; and

placing and routing a connection from said internal signal to said output pin using said logical netlist and said routing netlist, thus producing a modified routing netlist; whereby said connection introduces an additional delay from said internal signal to said output pin due to said registers.

21. A method as recited in claim 20 further comprising:

outputting said modified routing netlist into a programming output file (POF) in a form suitable for programming; and

downloading said POF onto a PLD, whereby said electronic design may be tested and debugged on said PLD by viewing said internal signal at said output pin.

22. A method as recited in claim 20 wherein the recited steps of claim 20 are performed in a system by an electronic design automation (EDA) tool.

23. A method as recited in claim 20 wherein said electronic design is intended for a programmable logic device (PLD), a chip, or a circuit board.

24. A method as recited in claim 20 further comprising:

performing a full compile of said electronic design prior to said incremental recompile, wherein said incremental recompilation does not disturb the placement and routing resulting from said full compile.

25. A method as recited in claim 20 wherein the quantity of said number of registers created is determined by user input.

26. A method as recited in claim 20 wherein the quantity of said number of registers created is determined by software that automatically calculates said quantity.
27. A method as recited in claim 20 further comprising:

connecting a clock signal to be used to trigger said registers in said logical netlist.
28. A method as recited in claim 20 wherein said registers are flip flops, latches or more complex registers.
29. A method of debugging an electronic design comprising:

selecting an internal signal of said electronic design to be viewed;

selecting an output pin of said electronic design to which it is desired to route said internal signal;

choosing a quantity of registers to be inserted between a source of said internal signal and said output pin;

executing a compile of said electronic design including compiling a routing from said source internal signal to said output pin via said quantity of registers to produce a compiled electronic design; and

downloading said compiled electronic design onto a PLD, whereby PLD includes an additional delay from said internal signal source to said output pin due to said quantity of registers.
30. A method as recited in claim 29 wherein said compile is an incremental recompilation, said method further comprising:

performing a full compile of said electronic design prior to said incremental recompilation, wherein said incremental recompilation does not disturb the placement and routing resulting from said full compile.

31. A method as recited in claim 29 further comprising:

identifying a clock signal to be used to trigger said registers.

32. A method as recited in claim 29 wherein said registers are flip flops, latches or more complex registers.

33. A method as recited in claim 29 wherein the recited steps of claim 29 are performed in a system using an electronic design automation (EDA) tool.